

We claim:

1. A method for generating a device driver for a device in an information processing apparatus that executes an application, wherein said device is connected to
5 said information processing apparatus, said method comprising the steps of:

obtaining a first part of data to generate a device driver for said device; and

generating said device driver for said device by using said first part of data and a second part of data stored in a memory, wherein said memory stores said application and said second part of data.

10

2. A method according to claim 1, wherein said first part of data is device model dependent configuration data.

3. A method according to claim 2, wherein said device dependent configuration
15 data is color conversion data to convert RGB color-space to a native color-space of said device.

4. A method according to claim 1, wherein said second part of data is device model independent data.

20

5. A method according to claim 1, wherein said generating step configures said first part of data with said second part of data.

6. A method according to claim 1, wherein said application in said memory is an
25 unchangeable application.

7. A method according to claim 1, wherein said information processing apparatus is a game console.

5 8. A method according to claim 6, wherein said unchangeable application is a game application executed on a game console.

9. A method according to claim 1, further comprising the step of copying said first part of data into a memory card connected to said information processing apparatus.

10

10. A method according to claim 1, wherein said first part of data is obtained from a disc for a printer.

15

11. A method according to claim 1, wherein said part of data is obtained from a server over a network.

12. A method according to claim 1, wherein said part of data is obtained from a memory in a printer.

20

13. A method according to claim 1, further comprising the step of determining a model of a device to which said application is desired to interface with.

14. A method according to claim 13, wherein said model of said device is determined through reading an identification string from said device.

25

15. A method according to claim 13, wherein said generating step is executed after executing said determining step.

16. A method according to claim 13, wherein said determining step determines
5 whether said first part of data in a memory card matches said device, and upon determining that said first part of data in said memory card matches said device, loading said first part of data from said memory card.

17. A method according to claim 16, upon determining that said first part of data
10 in said memory card does not match said device, said determining step further determines whether said first part of data in a disk matches said device, and upon determining that said first part of data in said disk matches said device, loading said first part of data from said disk.

18. A method according to claim 1, wherein said memory is a disc for a game.

19. A method according to claim 1, wherein said device is connected to said information processing apparatus by means of a USB connection.

20. A method according to claim 1, wherein said obtaining step is executed in the information apparatus when a printing operation is required by a user.

21. An information processing apparatus for executing an application and for generating a device driver for a device connected to said apparatus, said apparatus
25 comprising:

a processor for obtaining a first part of data to generate a device driver for said device and generating said device driver for said device by using said first part of data and a second part of data stored in a memory, wherein said memory stores said application and said second part of data.

5

22. A computer program to be executed in an information processing apparatus for executing an application and for generating a device driver for a device connected to said apparatus, said computer program comprising:

code for obtaining a first part of data to generate a device driver for said device;

10 and

code for generating said device driver for said device by using said first part of data and a second part of data stored in a memory, wherein said memory stores said application and said second part of data.

15 23. A computer program according to claim 22, wherein said computer program is stored in a memory medium to be connected to said apparatus.

24. A method of providing forward compatibility of device driver code of an unchangeable application with a plurality of device models, wherein said application is not linked to other executable code, said method comprising the steps of:

including device model independent device driver code in said application;
determining a model of a device to which said application is desired to interface with;

reading model dependent configuration data for said model of said device; and
25 configuring said device driver code with said model dependent configuration data.

25. A method according to claim 24, wherein said model of said device is determined through reading an identification string from said device.

5 26. A method according to claim 24 or 25, wherein said unchangeable application is a game application executed on a game console.

27. A method according to claim 26, wherein said model dependent configuration data is read from a memory card of said game console.

10

28. An information processing apparatus for providing forward compatibility of device driver code of an unchangeable application with a plurality of device models, wherein said application is not linked to other executable code and said device driver code is device model independent, said apparatus comprising:

15 storage means for storing model dependent configuration data for a plurality of devices;

means for determining a model of a device to which said application is desired to interface with;

20 data reading means for reading model dependent configuration data of said model of said device; and

means for configuring said device driver code with said model dependent configuration data.

29. Apparatus according to claim 28, wherein said device is connected to said
25 apparatus by means of a USB connection.

30. Apparatus according to claim 28 or 29, wherein said model of said device is determined through reading an identification string from said device.

31. Apparatus according to any one of claims 28, wherein said unchangeable application is a game application and said apparatus is a game console.

32. Apparatus according to claim 31, wherein said storage means is a memory card of said game console.

33. A computer program product, carried on a storage medium, for providing forward compatibility with a plurality of device models, wherein said computer program product is unchangeable and not linked to other executable code, said computer program product comprising:

an application;

model independent device driver code;

code for determining a model of a device to which said application is desired to interface with;

code for reading model dependent configuration data for said model of said device; and

code for configuring said model independent device driver code with said model dependent configuration data.